Scalable analysis and visualization of high-dimensional astronomical data sets

Ferdosi, Bilkis Jamal

This chapter is an extended version (submitted to the journal **Information Visualization**, invited paper) of: B. J. Ferdosi, H. Buddelmeijer, S. Trager, M. H. F. Wilkinson, J. B. T. M. Roerdink. Finding and visualizing relevant subspaces for clustering high-dimensional astronomical data using connected morphological operators. *IEEE Conference on Visual Analytics Science and Technology (IEEE VAST) in Salt Lake City, USA*, pp. 35-42, October 2010.

Chapter 3

Finding and Visualizing Relevant Subspaces for Clustering High-Dimensional Data Using Connected Morphological Operators

Abstract

Data sets in many scientific areas are growing to enormous sizes. For example, modern astronomical surveys provide not only image data but also catalogues of millions of objects (stars, galaxies), each object with hundreds of associated parameters. Gene expression experiments produce data about the complete genome of an organism under different conditions and at a sequence of time points. Exploration of such very high-dimensional data spaces poses a huge challenge. Subspace clustering is one among several approaches which have been proposed for this purpose in recent years. However, many clustering algorithms require the user to set a large number of parameters without any guidelines. Some methods also do not provide a concise summary of the datasets, or, if they do, they lack additional important information such as the number of clusters present or the significance of the clusters.

In this chapter, we propose a method for ranking subspaces for clustering which overcomes many of the above limitations. First we carry out a transformation from parametric space to discrete image space where the data are represented by a grid-based density field that also provides visual support for the analysis of the important subspaces. Then we apply so-called connected morphological operators on this density field. Clusters in subspaces correspond to high-intensity regions in the density image. The importance of a cluster is measured by a new quality criterion based on the dynamics of local maxima of the density. Connected operators are able to extract such regions with an indication of the number of clusters present. The subspaces are visualized during computation of the quality measure, so that the user can interact with the system to improve the results. In the result stage, we use three visualization toolkits linked within a graphical user interface so that the user can perform an in-depth exploration of the ranked subspaces. Evaluation based on synthetic as well as real astronomical and gene expression datasets demonstrates the power of the new method. We

recover various known relations directly from the data with little or no a priori *assumptions. Hence, our method holds good prospects for discovering new relations as well.*

3.1 Introduction

Data sets in many scientific areas are growing to enormous sizes. For example, modern astronomical surveys provide not only image data but also catalogues of millions of objects (stars, galaxies), each object with hundreds of associated parameters. In genomics, DNA microarrays are used to measure the expression levels of thousands of genes simultaneously. In addition, the gene expressions are measured as a function of time or under different experimental conditions, leading to a very large amount of high-dimensional data.

A main line of research is geared toward investigating multidimensional and multiscale patterns in the data. For example, an important task for data analysis in astronomical research is to explore the relation between galaxy morphology (i.e., the spatial distribution of objects) and the parameters associated to the objects which characterize the stellar environment. In genomics, one is interested to explore how cellular processes and functions are regulated by the complex interactions between large numbers of genes, proteins and metabolites. Although the high data rates required for acquisition, processing and populating the databases in these fields are well under control and are supported by dedicated project teams and software pipelines, we need to develop new approaches for extracting, analyzing and visualizing relevant information out of the flood of high-dimensional data.

Exploration of very high-dimensional information spaces poses a huge challenge. On the one hand, the techniques should cope with enormous amounts of data in a highly automated fashion, and be scalable to ensure that the newly developed methods remain usable while the data catalogues increase in size. On the other hand, the approach should allow the observer to participate in the analysis by using interactive visualization combined with the human perceptive and analytical power. This is especially true as the goal is to find "unexpected" phenomena in the data, for which by definition no *a priori* description is available, thus precluding the possibility of fully automated search.

Combining data mining approaches with visualization can enable users to explore such large datasets. Clustering is a well known data mining task that helps to discover natural structures in a dataset (Kriegel *et al.* 2009). Due to the exploratory nature of the task, full dimensional clustering techniques cannot help much. Clusters may exist in different subspaces that may indicate different relations among particular subsets of dimensions. Subspace clustering is an approach that can be applied for this purpose. Subspace clustering is the process of finding clusters in subspaces of the full feature space, either directly (Agrawal *et al.* 1998) or by identifying relevant subspaces for (later) clustering based on some quality criteria (Baumgartner *et al.* 2004).

In this chapter, which is an extended version of Ferdosi *et al.* (2010), we propose an approach to find relevant subspaces which is strongly tied to morphological properties of object distributions. Therefore, we apply techniques from the field of mathematical morphology, which was developed to describe image operators for enhancement, segmentation and extraction of shape information from digital images (Serra 1982, Heijmans 1994). In contrast to traditional linear

image processing, the morphological image operators focus on the *geometrical* content of images and are nonlinear, and many efficient algorithms are available for binary and grey scale images.

The main steps of our approach can be summarized as follows. First we carry out a transformation from the parametric space of the objects (galaxies, genes, etc.) to a discrete image space where the data are represented by a density field. This transformation is obtained by using grid-based density estimation. Local maxima in the grid density profile can be indicators of clusters/outliers in the dataset. Next we determine for each local maximum of the density field whether it represents a relevant subspace by applying quality criteria based upon the notion of *dynamics* (Bertrand 2007), which indicates the significance of a local maximum, see section 3.3.4.

The search for modes/local maxima is done on the so-called *Max-tree representation* of the density image. Such a representation is used in mathematical morphology to implement an important class of morphological operations known as *connected operators* (Salembier and Wilkinson 2009, Salembier *et al.* 1998). The main property of connected operators is that they do not process individual data points, but entire connected components at each grey level. Such components are either kept or completely removed by the operator. Therefore, such operators can be used to perform filtering based on various shape and size attributes. More information on connected operators is provided in section 3.3.3. For subspaces of dimension higher than three we apply principal component analysis (PCA) and use the first three principal components for subspace ranking. The main reason for using PCA is that for higher dimensions the current Max-tree implementation becomes prohibitive in terms of computing time and memory use.

Along with the quality measure and ranking of the subspaces we provide quantitative information such as the number of clusters present, degree of separation, size and shape of the clusters, etc. Note that our method does not perform the actual clustering itself, i.e., it does not assign points to clusters. For this purpose, existing clustering algorithms (such as k-means) may be used.

Visualization plays an important role in our approach. The subspaces are visualized during computation of the quality measure, so that the user can interact with the system to improve the results. In the result stage, we use an interactive tree visualization providing all sorts of statistics about each subspace along with the ranking. We also link three visualization toolkits within a graphical user interface so that the user can perform an in-depth exploration of the ranked subspaces.

Our main contributions can be summarized as follows:

- We introduce the use of connected morphological operators to analyze grid density profiles of subspaces of parameter space;
- We propose a subspace quality criterion based on the dynamics of maxima found in the density profile;
- Linked visualizations are used to support the user in the exploration of the subspaces.

The remainder of the chapteris organized as follows. Related work is discussed in section 3.2. Section 3.3 then describes the working principle of our subspace finding method, including the background on density estimation, connected morphological operators and the concept of dynamics. Our interactive visual subspace exploration system is described in section 3.4. We

present the experimental results of the method in section 3.5. Section 3.7 gives a summary along with plans for future work.

3.2 Related Work

3.2.1 Subspace Clustering and Ranking

A well known method to rank subspaces for clustering is the SURFING ("SUbspace Relevant For clusterING") method (Baumgartner *et al.* 2004). It belongs to the class of methods that only compute interesting subspaces rather than final subspace clusters (Kriegel *et al.* 2009). Relevance of a (sub)space is measured through a quality criterion based on a hierarchical clustering structure of subspaces. The method is based on the idea that subspaces with clusters of different densities and noise will show significant variation in k-nearest neighbor distances compared to subspaces with a uniform distribution. The quality of a subspace is determined as a function of differences of distances to the mean distance of the objects. The precise definition of quality is as follows. Let *DB* be a set of feature vectors

$$quality(S) = \begin{cases} 0 & \text{if } Below_S = 0 \\ \frac{diff_{\mu_S}}{|Below_S| . \mu_S} & \text{otherwise} \end{cases} \tag{3.1}$$

where

$$diff_{\mu_S} = \frac{1}{2} \sum_{0 \in DB} |\mu_S - nn - Dist_k^S(o)| \tag{3.2}$$

where μ_S is the mean of k-nearest neighbor distances of the objects, and for an object o, $nn - Dist_k^S(o) = \max\{distance(o_s, p_s) | p \in NN_k^S(o)\}$. SURFING can be very helpful where in-depth knowledge of the spaces can be traded against high processing speed, e.g., in web services. However, this method only gives a qualitative ranking of the subspaces without any quantitative information such as the number, size, shape or separation of the clusters.

In Kailing *et al.* (2003) another density-based subspace selection method called RIS ("Ranking Interesting Subspaces") is proposed. Relevance is computed as a function of core objects (i.e., objects inside a cluster, Sander *et al.* (1998)). Subspaces that contain no core objects are pruned in a bottom-up way. The performance of this method largely depends on proper tuning of a large number of parameters, which is sometimes hard to achieve. It also uses a global density threshold for subspaces with different dimensionality that can prevent the method from finding a proper clustering structure existing in different subspaces of varying dimensionality.

There are other methods like CLIQUE (CLustering In QUEst) (Agrawal *et al.* 1998), ENCLUS (ENtropy-based CLUStering) (Cheng *et al.* 1999), DOC (Density-based Optimal projective Clustering) (Procopiuc *et al.* 2002), or PROCLUS (PROjected CLUStering) (Aggarwal *et al.* 1999) that perform direct cluster computation in subspaces. CLIQUE first finds candidate subspaces by computing a histogram in each of the dimensions and selecting the dense ones. Then clusters are computed in the subspaces that are selected by a criterion that satisfies a downward closure (or monotonicity) property (Kriegel *et al.* 2009). This criterion says the following:

if a subspace S contains a cluster, then any subspace $T \subseteq S$ must also contain a cluster; see also Kriegel *et al.* (2009). Pruning subspaces is done by the MDL (Minimal Description Length) principle. However, CLIQUE provides no information on the subspaces in which the whole dataset clusters best. Top-down pruning can miss many small but significant clusters. It also is difficult to find a proper tuning of parameters for different datasets.

ENCLUS is based on the CLIQUE algorithm but instead of density it uses entropy to find the candidate subspaces. This method also has all the advantages and disadvantages of CLIQUE. DOC proposes a mathematical definition of an optimal projective cluster in subspaces. Density is measured with a fixed-width hypercube. However, this may not be appropriate for varying density of different subspaces. Finding proper values for a large number of parameters is another problem of this method. PROCLUS is one of the clustering oriented approaches that focus on the clustering result by directly specifying objective functions, like the number of clusters to be detected or the average dimensionality of the clusters. Both parameters are hard to set because in most of the cases they are unknown. Taking a fixed dimensionality of the subspaces is not appropriate either, since clusters may be present in various combinations of dimensions.

3.2.2 Visualization

Integration of visualization in the subspace ranking and clustering process seems to be a less explored area. Assent *et al.* (2007) proposed a visualization paradigm to present and explore clusters from subspace clustering. Using multidimensional scaling (MDS) they present information like (dis)similarity, overlap, size, dimensionality etc., of the resulting clusters. They provide an aid to parameter tuning in terms of *bracketing*, a technique originating from photography. A matrix representation is used to visualize the grouping of clusters. However, these visualization approaches are about presentation of clustering results, but do not aid in exploring individual subspaces, our goal in this Chapter.

3.3 Searching Relevant Subspaces for Clustering

3.3.1 Overview of the Method

Let us denote by $DATA$ a set of N data points (rows) with d dimensions (columns), i.e., $DATA \subseteq \mathbb{R}^d$. Let $A = \{a_1, ..., a_d\}$ be the set of all attributes a_i of $DATA$. A subspace in $DATA$ is a set S with $S \subseteq A$. We define a subspace as *relevant* if it does not contain uniform noise or only a single Gaussian distribution spread over the whole attribute range. Therefore, the emphasis is given on multimodality of the density where each mode is indicative of a cluster. The degree of relevance is determined in terms of significance and separability of each mode (indicator of a cluster) in the multimodal distribution.

We search for the modes and determine their significance and separability in grey level image space, whereas most of the traditional subspace clustering methods work in parametric space. The motivation for working in discrete image space is that the number of grid points can be chosen to match the desired grid resolution, while the number of data points may grow very

large. This representation facilitates the analysis of the subspaces because of the structured representation using the Max-tree. Also, it allows an easy integration of the visualization of the density field.

Therefore, a transformation of parametric space to image space is required. This transformation is obtained by using grid-based density estimation, as described in section 3.3.2. Thus modes in the distribution are transformed into high-intensity peaks (local maxima) in the density image. However, densities produced by the estimator have continuous values and thus the densities have been discretized to obtain discrete gray levels in images used in the later stages.

The search for modes/local maxima is done on the Max-tree representation of the density image, see section 3.3.3. Each node of the Max-tree with a certain grey level contains all the connected components at that level. Connected components are obtained using neighborhood relationships in the grid. The root of the tree contains the connected components with lowest intensity and the leaves contain the connected components with highest intensity. Therefore, counting the number of leaves gives us the number of clusters.

The significance and separability of modes is determined using the concept of relative dynamics as described in section 3.3.4. Significant and well-separated modes will have higher relative dynamics compared to overlapping clusters. To derive a quality criterion for subspaces we use the number of modes (number of leaves in the Max-tree) and their relative dynamics, see section 3.3.5.

3.3.2 Density Estimation

Density estimation is one of the techniques of choice to uncover structure in point-set data (Silverman 1986). We estimate the density of each subspace by a fast and scalable modification from (Wilkinson and Meijer 1995) of the adaptive kernel density estimation method of Breiman *et al.* (Breiman *et al.* 1977). A brief discussion of the method is presented here; for details please see chapter 2.

For a data sample of N points with position vectors $\vec{r}_i = (r_{1,i}, r_{2,i}, \ldots, r_{d,i}) \in \mathbb{R}^d, (i = 1, ..., N)$, the adaptive kernel density estimate $\hat{p}(\vec{r})$ is given by:

$$\hat{p}(\vec{r}) = \frac{1}{N} \sum_{i=1}^{N} (h_1 \ldots h_d)^{-1} \lambda_i^{-d} K_e \left(\frac{r_1 - r_{1,i}}{h_1 \lambda_i}, \ldots, \frac{r_d - r_{d,i}}{h_d \lambda_i} \right) \tag{3.3}$$

Here the local bandwidth is the product of a window size h_ℓ depending on the coordinate direction $\ell = 1, 2 \ldots, d$ and a local bandwidth parameter λ_i for each data point i. In this formula K_e is the Epanechnikov kernel defined as

$$K_e(\vec{t}) = \begin{cases} \frac{d+2}{2V_d}(1 - \vec{t}.\vec{t}) & \text{if } \vec{t}.\vec{t} < 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

in which V_d is the volume of the unit sphere in d-dimensional space. In this model we have to choose the local bandwidth parameters λ_i in such a way that in low-density regions λ_i will be large and the kernel will spread out; in high-density regions the opposite should occur.

The density estimation proceeds in two phases.

Phase 1. Use a percentile of the data to compute an optimal pilot window width h_ℓ^{opt} in each of the coordinate directions:

$$h_\ell^{opt} = \frac{P_{80}(\ell) - P_{20}(\ell)}{\log N}, \quad \ell = 1, \ldots, d \tag{3.5}$$

where $P_{80}(\ell)$ and $P_{20}(\ell)$ are the 80$^{\text{th}}$ and 20$^{\text{th}}$ percentile of the data points in dimension ℓ. Define a pilot density $\hat{p}_{pilot}$ using $\lambda_i = 1$ for all $i = 1, 2 \ldots, N$ and $h_\ell = h_\ell^{opt}$ in formula (3.3).

Phase 2. From the pilot density $\hat{p}_{pilot}$ compute the local bandwidth parameters λ_i by

$$\lambda_i = \left(\frac{\hat{p}_{pilot}(\vec{r}_i)}{g} \right)^{-\alpha}. \tag{3.6}$$

Here g is the geometric mean of the pilot densities and $\alpha = 1/d$ is the sensitivity parameter. The final density estimate is given by formula (3.3) once again, but now with λ_i given by (3.6) with $h_\ell = h_\ell^{opt}$.

The Epanechnikov kernel has finite support so that computation time is reduced significantly. The density is estimated on a Cartesian grid, which includes all data points. The method is computationally effective: the complexity is $O(N)$; the computation time will increase for larger values of the smoothing parameter. Because of its grid structure the computed density can be visualized immediately by standard volume rendering techniques for $d \leq 3$. In our method a fundamental use of the grid structure is to obtain a neighborhood definition for computing connected components in the density field. Note that the grid must be finer than the smallest window size.

3.3.3 Connected Morphological Operators

A connected operator can extract and filter connected components known as flat zones, i.e., constant intensity regions, where connectivity is defined on the digital grid. Connected operators create a hierarchy of flat-zone partitions with an ordering relation. The Max-tree data structure can be used to implement such a hierarchy (Salembier and Serra 1995, Salembier *et al.* 1998).

Consider a digital image I on a domain $D \subseteq \mathbb{Z}_n$ with 2-adjacency for $n = 1$, 4 or 8-adjacency for $n = 2$, and 6 or 26-adjacency for $n = 3$. A set $X \subseteq D$ is connected if each pair (p, q) of points in X can be joined by a path $(p_0, p_1, \ldots, p_{\ell-1}, p_\ell)$ such that $p_0 = p$, $p_\ell = q$ and (p_i, p_{i+1}) are neighbors $\forall i \in [0, \ell)$. A connected component of X is a connected subset $C(X)$ of X which is maximal. A flat zone at grey level h of I is a connected component of the level set $X_h(I) = \{p \in D | I(p) = h\}$.

Max-tree representation. In the Max-tree representation of an image the root corresponds to the flat zone with lowest intensity and leaves contain the flat zones with highest intensity (Salembier and Serra 1995, Salembier *et al.* 1998). Local maxima in the image correspond to connected sets of constant value which are separated from other local maxima by local minima. An illustration is given in Fig. 3.1. In the left image of this figure there are three well-separated clusters with varying intensity. In the right image the corresponding Max-tree representation is shown. The

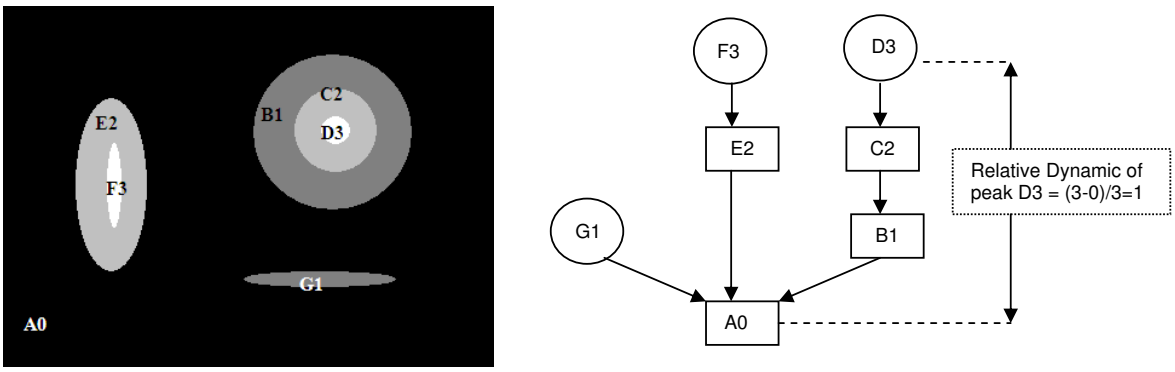

Figure 3.1. *Left: grey level image that contains three connected components with varying intensity. Right: Max-tree representation of the left image. Max-tree node A0 represents the background, and the other connected components are indicated by B to G along with their grey values. The relative dynamics of peak D3 is also indicated.*

Max-tree node A0 represents the background. As there are two flat zones with grey level 1 and one with grey level 2, the root has two child nodes (B1, G1) at level 1 and one child node (E2) at level 2. Each of the flat zones can be a leaf or have children. Flat zones with maximum intensities are in the leaves (G1, F3, D3). The Max-tree is a rooted tree, thus every node has a pointer to its parent. The Max-tree is constructed with a recursive flood filling with a FIFO queue to process the pixels/voxels in the correct order.

Each node in the Max-tree can contain several size or shape attributes that can be calculated incrementally during the tree construction. Some example attributes are *Size*, i.e., the area A of the flat zone as defined by the number of pixels in that zone, or the scale invariant shape attribute defined by M/A^2, i.e., the ratio of moment of inertia M and the square of the area A. The Max-tree along with the attributes can be computed in a time which is linear in the number of pixels.

3.3.4 Dynamics

In image analysis the concept of "dynamics" is used as a measure of contrast. It can be used to rank the local maxima of an image (Bertrand 2007). In the image processing literature the dynamics of a minimum m with height H_m is generally defined with the concept of flooding. After placing a unique flooding source at m, if the height of the flood is H_1 when for the first time during flooding a catchment basin with a deeper minimum than m is reached then the dynamics of m is defined as $H_1 - H_m$. The dynamics of a regional maximum m is defined analogously, by considering the path of minimal altitude linking m for the first time to another maximum of higher altitude than m (Vachier and Vincent 1995). However, in our work we used a modified definition of dynamics: instead of considering "another maximum of higher altitude than m" we only consider "another maximum". In the Max-tree the local maxima are in the leaves. Therefore, the dynamics of a local maximum is the difference between the intensity value of the corresponding leaf and the intensity value of the first ancestor with multiple children that corresponds to the

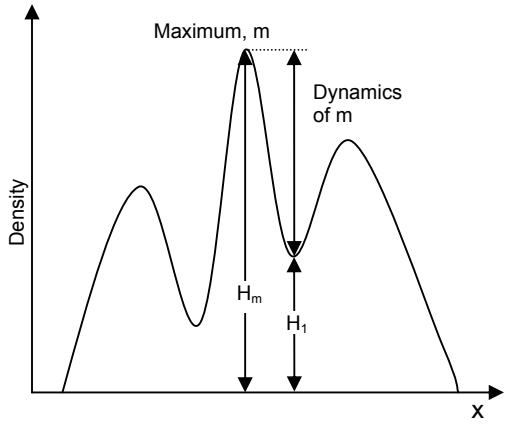

Figure 3.2. *Dynamics of a local maximum m.*

minimum that links m with another maximum (cf. Fig. 3.1). One problem with this definition is that a maximum with low amplitude can be treated as insignificant compared to a maximum with large amplitude. Therefore, we use *relative* dynamics so that all maxima are treated equally, i.e., when m is a local maximum its relative dynamics is defined by

$$RelativeDynamics(m) = (H_m - H_1)/H_m. \tag{3.7}$$

For the example of Fig. 3.1 this means that all the maxima have a relative dynamics of 1. Relative dynamics are also scale-invariant, because a linear scaling of the data space scales all the densities linearly as well.

The reason behind our modified definition of dynamics is that it allows us to identify the separability of the modes. For example, in figure 3.3 we show a schematic diagram of two overlapping modes and one well separated mode, their Max-tree representation, and the corresponding relative dynamics of the modes. The well separated mode received the relative dynamics of 1 and the overlapping modes received the relative dynamics of 0.25 in this example.

3.3.5 Subspace Quality Criterion

Let S be a subspace of the space A of attributes. The quality of S, denoted by *Quality*(S), is defined as follows

$$Quality(S) = \begin{cases} N_L^{-1} \sum_{i=1}^{N_L} RelativeDynamics(i) & \text{if } N_L > 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

where N_L is the number of leaves in the Max-tree. In this criterion the sum of the dynamics of all local maxima is normalized by the number of local maxima and thus the value of *Quality* ranges from 0 to 1. A subspace that contains modes/clusters with high dynamics will have a

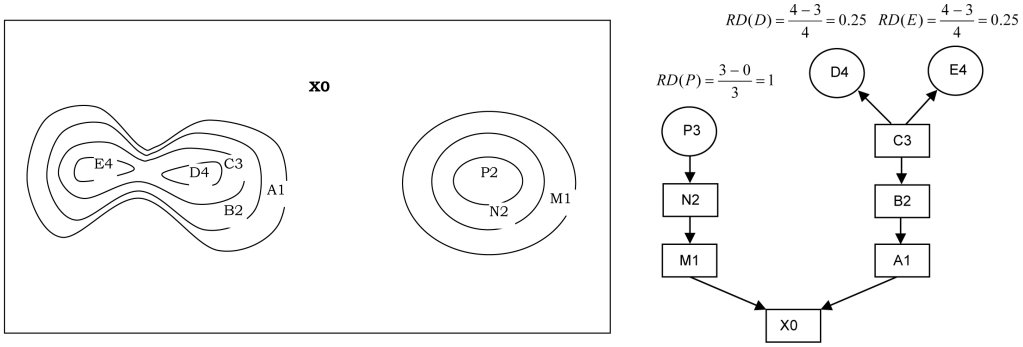

Figure 3.3. *Measuring separability of the modes using a modified definition of dynamics. (Left) schematic diagram of two overlapping modes and one well separated mode; (right) Max-tree representation of the modes and their relative dynamics (RD).*

higher value of *Quality* than a subspace with clusters of lower dynamics. A subspace as depicted in Fig. 3.1 will have a quality of 1 according to equation (3.8) because of the presence of three modes with dynamics of 1 each. Two important aspects of our quality criterion are: (*i*) the use of relative dynamics allows us to treat clusters with varying density equally; (*ii*) the quality criterion is unbiased in ranking subspaces with varying number of clusters because of the normalization by the number of leaves.

Note that in our method, subspaces with the same quality, but with varying numbers of clusters, get the same ranking and thus they will be grouped together in the rank list. However, along with the ranking, our method also provides information about the number of clusters that may be present. Therefore, it becomes possible for the user to choose the subspace of interest (with more/less clusters) from the group of subspaces with the same quality, unlike other methods where such grouping is not available.

3.3.6 Subspace Finding

The search for subspaces is performed in a bottom-up fashion, i.e., starting from one-dimensional subspaces, then moving to two-dimensional subspaces, etc. The process of finding relevant subspaces is summarized in the pseudo code of Algorithm 3.1. Up to dimension three the creation of the density image, Max-tree construction and computation of the quality index is done on the original feature space. For subspaces of dimension higher than three we apply PCA and use the first three principal components for subspace ranking. The main reason is that for higher dimensions the current Max-tree implementation becomes prohibitive in terms of computing time and memory use. Using PCA globally in the full dimensional feature space is open to criticism. However, in our approach we are using it in local feature spaces. Therefore, we can avoid the drawbacks of global usage of PCA. An added benefit of our choice to use the first three principal components of PCA is that we can use standard volume rendering to visualize the density fields.

Ranking and Pruning. Based on the quality of the subspaces we produce a ranking. Unlike SURFING we do not discard any of the subspaces in the one dimensional search. Discarding

spaces in such an early stage can reduce the search space dramatically but it also precludes the possibility of finding interesting relations in later stages that may arise with the combination of discarded 1-D subspaces. However, it is necessary to prune the subspaces because of their exponential growth. Therefore, we introduce pruning for 2-D and higher dimensions. We prune a subspace if it has a quality value less than a threshold value θ. From our study on several uniformly distributed spaces we found that they always have a quality value less than 0.1. Therefore, we set $\theta = 0.1$.

```
1:  DATA ← d-dimensional dataset;
2:  A={a_1.,..,a_d}; // attribute set
3:  n = 1;
4:  while n ≤ d do
5:     NrOfSpaces ← (d choose n);
6:     S_n ← set of n-dimensional subspaces S_{n,j}, j = 1,...,NrOfSpaces;
7:     for j = 1 to NrOfSpaces do
8:        if (n > 3) then
9:           S_{n,j} ← ComputePCA(S_{n,j});
10:       end if
11:       Den_{n,j} ← ComputeDensityField(S_{n,j});
12:       Visualize(Den_{n,j});
13:       WaitForInteraction;
14:       if (interaction) then
15:          Accept new smoothing parameter
16:          go to 11;
17:       else
18:          M_{n,j} ← CreateMaxTree(Den_{n,j});
19:          quality(S_{n,j}) ← ComputeQuality(M_{n,j});
20:       end if
21:    end for
22:    rank according to quality;
23:    //Pruning for n > 1
24:    if (n > 1 and quality(S_{n,j}) < θ) then
25:       remove S_{n,j};
26:    end if
27:    n ← n + 1;
28: end while
```

Algorithm 3.1: *SubspaceFinding*

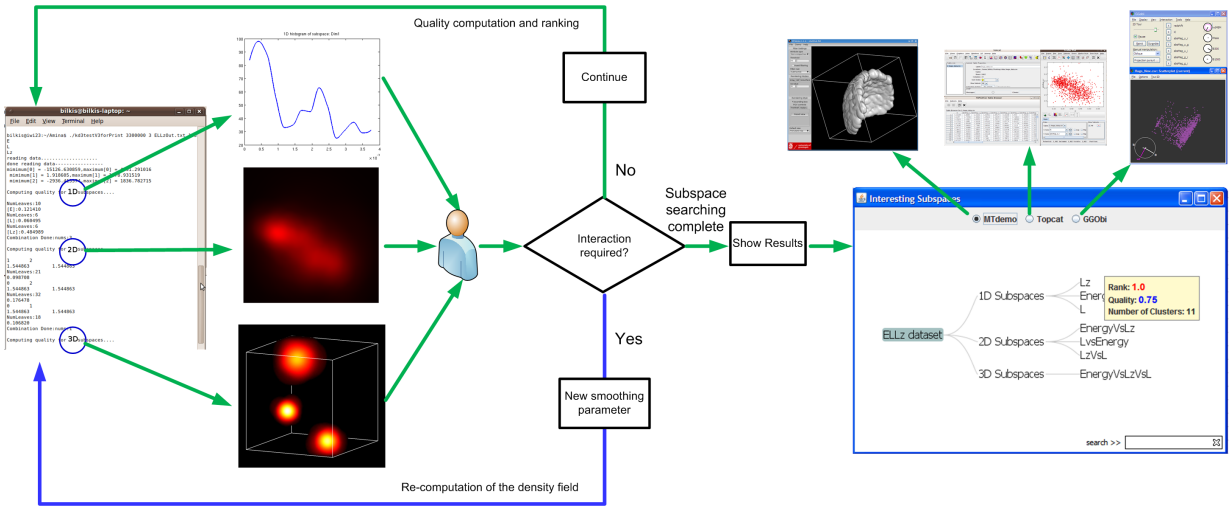

Figure 3.4. *Schematic diagram of our interactive search and exploration system.*

3.4 Interactive Visual Subspace Exploration

An overview of our subspace search and exploration system is given in Fig. 3.4. The left part of the figure shows the quality computation process. It is very important to choose a proper value for the smoothing parameter during density computation. Most of the current density-based approaches for subspace clustering and ranking try to find a proper parameter by trial and error, which is very cumbersome (Müller *et al.* 2009). Initially, we provide an automatic setting of the smoothing parameter as described in section 3.3.2. Most of the time this automatic selection works. However, it may produce an under-/ over-smoothed density field, which is best detected through visual inspection by the user. Therefore, in our method we visualize the density field with standard volume visualization for 3-D and higher dimensions. For 2-D we visualize it as an image and for 1-D the histogram of the point densities is used. If the users detect any over-/ under-smoothing they can interact with the system to give a new smoothing parameter value.

We represent the result of the relevant subspace finding method by a tree visualization (see right side of Fig. 3.4). The root represents the complete dataset, the next level contains d nodes where d is the dimensionality of the dataset. Each node contains a number of leaves, say m, where m is the number of relevant subspaces. If the mouse pointer hovers over a node an information box will appear with all the relevant information about that subspace. By clicking on the node a window will appear with a 1-D or 2-D density plot for dimension one and two, and a volume visualization of the density field for dimension three. For dimensions higher than three the density field of the first three principal components is visualized.

The tree can be panned (scrolled) to explore the branches. The user can also zoom in/out for better reading in case that the tree is large or too cluttered. We combine three visualization

tools with our interface. From the top panel the user can choose Topcat[1], GGobi[2] or MTdemo[3] to check if the subspaces are really relevant. Topcat is a well known table visualizer in the astronomical community that also has different plotting capabilities. It is quite competent in handling very large high-dimensional data. GGobi is also a well-known information visualization tool that provides several high-dimensional data visualization techniques. For volume visualization we use MTdemo, a Max-tree-based volume visualization tool presented by Westenberg *et al.* (2007). It renders the volume with three different rendering techniques, X-ray, Maximum Intensity Projection (MIP) and Isosurface. MTdemo is not only a volume visualization tool but also an attribute filtering tool. It allows the user to explore the volume by applying different shape preserving attribute filters.

In the overall workflow, the user would start with the subspace search, and interact until the result is satisfactory. Then the results are inspected by the tree visualization system described above. At this level, the subspace identification process can only be changed by restarting the process. However, in the future we will consider adding a feedback loop to the subspace search module by allowing the user to change the parameters of the subspace ranking process.

The amount of interaction will differ in the two phases. In the subspace ranking phase the smoothing parameter can be changed interactively. To make the subspaces comparable we normalized the units along each axis. Therefore, the scaling parameter should not vary for different subspaces of a particular dimension. Thus, inspecting one subspace per dimension should be sufficient. Still the number of inspections per dimension will depend on how often the smoothing parameter is changed, which can vary from user to user. Once the subspace ranking is complete, the number of inspections will be limited, as the subspaces are ranked by relevance. Usually, domain users have concrete hypotheses they want to verify and hence they will only explore the most relevant subspaces.

3.5 Experiments and Results

We compare the ranking performance of our method with SURFING, and the performance in finding the number of clusters with CLIQUE, as SURFING does not provide the latter information. As the source code was not available to us we used our own implementation of SURFING following the algorithm presented in Baumgartner *et al.* (2004). For CLIQUE we used the ELKI[4] platform (Achtert *et al.* 2008). We used several synthetic datasets, two astronomical datasets and one gene expression dataset for this purpose. Reported timings were obtained with an AMD athlon 64 X2 Dual core processor 5200+, 2.6 GHz and memory 1.94 GB.

[1] http://www.star.bris.ac.uk/~mbt/topcat

[2] http://www.ggobi.org

[3] http://www.cs.rug.nl/~michael/MTdemo

[4] http://www.dbs.ifi.lmu.de/research/KDD/ELKI

3.5.1 Synthetic Data

We created several synthetic datasets with varying numbers of clusters of varying dimensionality with different noise levels. Clusters were created as multimodal Gaussian distributions with different mean and variance. Depending on the value of the variance we created clusters with varying density. Then impulse noise was inserted uniformly, where the number of noise points varied from 0% to 10% of the number of points in the clusters. In Table 3.1 a brief summary of the synthetic datasets can be found.

The field "data dimension" indicates the dimensionality of the dataset. "Number of clusters" indicates the number of Gaussian clusters present in the dataset and "Cluster dimensions" indicates the dimensionality of the Gaussian clusters. For example in dataset 2, the dimensionality of the dataset is 12 ($d_1, d_2, \ldots, d_{12}$), and there are four 3D Gaussian clusters (in d_2, d_4, and d_6) with 10% uniformly distributed noise added and two 6D Gaussian clusters (in $d_7 - d_{12}$) present in the datasets without noise; the remaining dimensions (d_1, d_3, and d_5) of the dataset contain uniformly distributed random noise.

Table 3.1. *Synthetic datasets*

Dataset	Data Dimension	Number of clusters	Cluster dimensions
1	16	2	3
2	12	4,2	3,6
3	15	3	4
4	22	5	5
5	12	3	2

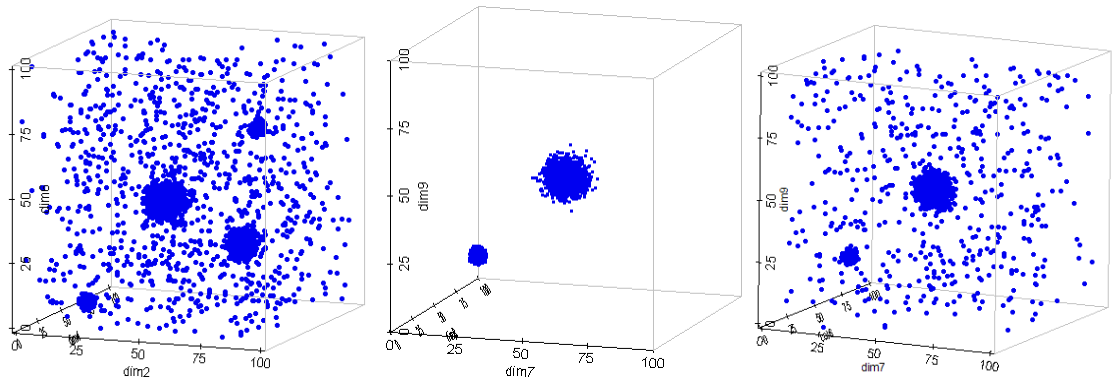

Figure 3.5. *Scatter plot of subspace-A (left), subspace-B from (middle), subspace-C with noise added to subspace-B (right).*

Performance for synthetic data. The performance of our method for synthetic datasets is satisfactory. It ranks subspaces with clusters always high in the list irrespective of the noise levels. It ranks subspace-A,B,C (Fig. 3.5) as equally relevant since they all get a quality value of 1. Our method puts emphasis not only on the number of clusters but also on their separability. Subspaces that have well separated clusters always come up high in our ranking. It also can indicate

the number of clusters properly in most of the cases. Sometimes fewer clusters than present are reported if there are overlapping clusters with one very high density and another with very low density.

SURFING puts most of the subspaces which do contain clusters higher in the ranking. However, noise-free cluster structures are penalized compared to clusters with noise in this method, see 'subspace-A' and 'subspace-B' in Fig. 3.5, left and middle, respectively. In subspace-A there are four Gaussian clusters of varying density with uniformly distributed noise that covers all clusters. In subspace-B there are two clusters without noise. The SURFING method put subspace-A in the top ranking as expected. However, it ranked subspace-B only as the 20^{th} relevant subspace in the list. Note that this result was obtained in spite of the fact that we introduced 1% of additional random points when calculating the SURFING quality measure, as recommended by Baumgartner *et al.* (2004). The motivation for adding a small percentage of random points is that SURFING's quality measure is based on the difference between k-nearest neighbor distances and mean distances. Hence, if a subspace has multiple clusters with the same density and without noise, it would get the same quality value as uniformly distributed points and thus remain lower in the ranking. By contrast, for cases where the clusters are fully covered by noise, as in Fig. 3.5 right, we found that SURFING does rank the subspace equally high as subspace-A in the list of relevant subspaces.

CLIQUE missed some of the clusters and sometimes detected unclear clusters. The main difficulty of CLIQUE is the need to find proper parameter sets that work for individual datasets.

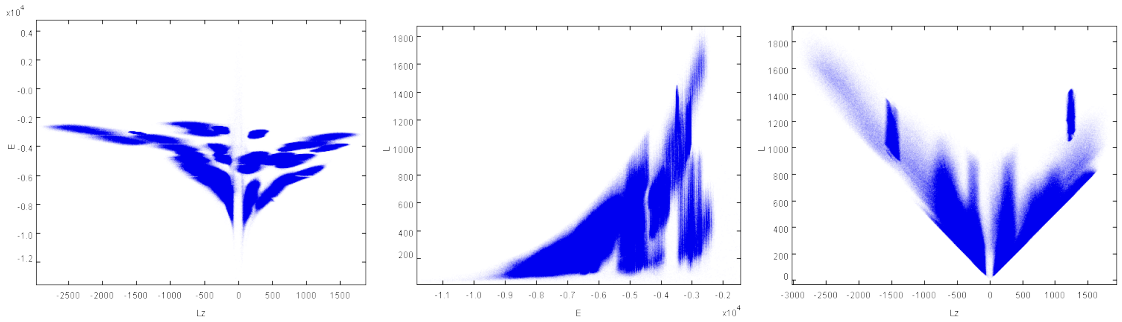

Figure 3.6. *Scatter plot of* Galactic stellar halo *dataset. E vs L_z (left), E vs L (middle), L vs L_z (right).*

For the synthetic datasets we checked whether the use of PCA caused any high (i.e., larger than 3) dimensional clusters to be missed. We found that this only occurred in dataset 3, where one of the three 4D clusters was missed. In dataset 4 all the five 5D clusters and in dataset 2 both 6D clusters were indicated.

3.5.2 Astronomical Data

We used two astronomical datasets. The first one is the *Galactic stellar halo* (roughly spherical outskirts of a galaxy) dataset, which is the result of a simulation. The second is a galaxy sample from SDSS (*S*loan *D*igital *S*ky *S*urvey), cf. `http://www.sdss.org`.

Table 3.2. *Comparison of methods on* Galactic stellar halo *dataset*

Method	**Ranking**		**Nr. clusters indicated**
	1-D	2-D	
Our method	L_z	$E-L_z$	31
	E	$E-L$	
	L	$L-L_z$	
SURFING	E	$L-L_z$	n.a.
	L	$E-L_z$	
	L_z	$E-L$	
CLIQUE	(in terms of coverage)	(in terms of coverage)	15
	E	$L-L_z$	
	L	$E-L_z$	
	L_z	-	

Galactic stellar halo dataset. This consists of 33 satellite galaxies each of them represented by a collection of 10^5 particles. It has been assumed that the whole stellar halo is the superposition of several disrupted satellite galaxies which fell into the Milky Way about 10^{10} years ago. It is possible to isolate remnants of satellite galaxies since stars in galaxies harbor unique clues of the assembly history of galaxies. The dataset contains three phase space coordinates, i.e., energy E, total angular momentum L and the z-component of angular momentum L_z. These three parameters are approximately conserved quantities that do not evolve much. Among them only L_z is fully conserved and thus should play the most important role in finding clusters. According to Helmi and de Zeeuw (Helmi and Zeeuw 2000) most structure is visible in the 2-D subspace $E-L_z$. With current approaches such as the *friends of friends* algorithm (Efstathiou *et al.* 1988) only 50 percent of the clusters have been recovered so far.

We applied all the methods to the *Galactic stellar halo* dataset. The results are shown in Table 3.2. Our method has the best performance in correctly ranking the parameters and also in indicating the maximum number of clusters. The fact that our method is able to detect 31 out of 33 clusters is a great advance compared to the performance of state of the art astronomical methods which reach only half of this (Helmi and Zeeuw 2000). However, it may be worth mentioning that in this work we do not perform the clustering itself. Therefore, we cannot ensure a one-to-one association of the particles in the cluster with the original satellites. In the future, we plan to obtain such an association and find out the clustering performance.

The ranking of our method is understandable if we look at the scatter plot of the 2-D spaces, see Fig. 3.6. The highest ranked 2-D subspace is $E-L_z$, which indeed has the largest number of clusters. However, CLIQUE's ranking in terms of coverage does not correspond to existing astronomical knowledge about the parameters. For example according to CLIQUE L_z has clusters with the least coverage of the dataset. However, according to Helmi and de Zeeuw L_z should contain more clustering information than the other parameters, as it is the most conserved quantity. Ranking of the 2-D subspaces is reasonable, although the method did not find any cluster in subspace $E-L$. CLIQUE found that subspace L-L_z has the clusters with highest coverage. It

can be inferred that this subspace has less clusters with large size. CLIQUE found less than half of the clusters present.

The ranking of SURFING for this dataset corresponds to the results of CLIQUE. In 1-D subspaces energy E is in the top ranking, followed by L and L_z respectively. In 2-D subspaces L-L_z is indicated as the most relevant subspace. If we look at the scatter plot of Fig. 3.6 it is evident that the L-L_z and E-L subspaces have more variations in their point distribution in space. On the other hand, the E-L_z space has more clumped structures when compared to the other two subspaces. This may indicate the weakness of measuring relevance only based on variation in point distances.

Galaxy sample from SDSS. This data set contains mainly photometric information of galaxies in the Northern Galactic Cap of SDSS Data Release 7 (Adelman-McCarthy *et al.* 2007). There are 32228 galaxies with 15 attributes in total present in this dataset, see Table 3.3.

The sample is limited to a spectroscopically measured distance range of 418 to 460 Mpc (1Mpc $\approx 3 * 10^{19}$km) to control distance related effects. It is difficult to compare galaxies at different distances: they are observed at different cosmological times and with different recessional velocities. An upper r-band Petrosian (Petrosian 1976) magnitude of 17.7 is imposed, to ensure a volume-complete sample for the quantification of the environment around the galaxies.

Two of the attributes, i.e., *magnitude* and *color*, are important in optical astronomy and need some elaboration. Magnitude refers to the luminosity of a galaxy in a specific wavelength band of the electromagnetic spectrum. Higher magnitude values correspond to fainter objects, lower values to brighter objects. In the galaxy dataset we used extinction-corrected model magnitudes: *dered_r* is the magnitude of galaxies measured in the r-band (around a wavelength of 6280 Å). The colors of a galaxy are defined as the differences between magnitudes in two different bands (Zeilik and Gregory 1998) such that the higher the color value the redder the galaxies are. In this dataset 10 different colors are used, such as *u-r, u-g*, etc. This allows us to study the influence of different colors in finding galaxy properties. The (inverse) concentration index is a measure of the light distribution of a galaxy.

In our performance measurement on the *SDSS galaxy sample* dataset we recover several well known relations of galaxy properties. In color vs magnitude a bi-modal distribution of red and blue galaxies can be observed (Baldry *et al.* 2004). Red galaxies are elliptical galaxies with mostly old stars and blue galaxies are spiral galaxies with mostly young stars. In the color vs inverse concentration index relation, this galaxy bimodality can also be observed (Baldry *et al.* 2006).

In 1D, the galaxy bimodality can be observed in the histogram of colors. Current astronomical research shows that this can best be seen in *color(u-r)*. This is confirmed by our method for ranking for 1-D subspaces, where *color(u-r)* is ranked first. On the other hand, SURFING ranked *logMass* highest. If we compare the histogram of these two subspaces (see Fig. 3.7) it is clear that *logMass* is not relevant in terms of clustering. On the other hand the *color(u-r)* histogram confirms the astronomical findings.

When we search in 2-D subspaces the combination *dered_r* vs *color(u-r)* is the first subspace among color vs magnitude combinations that appears in the ranking of our method. On the other hand SURFING ranks *dered_r* vs *color(r-i)* first. We can see a clear bimodality in the density

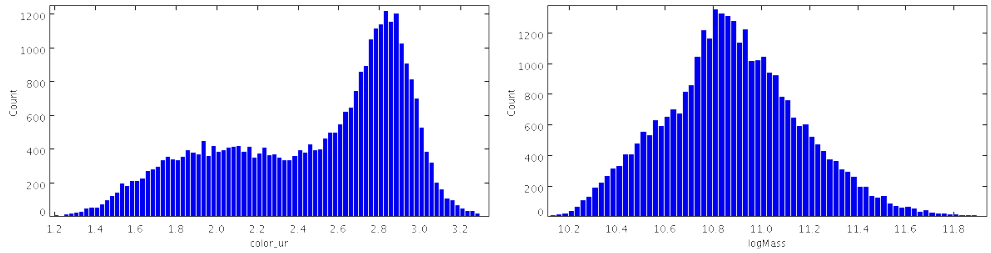

Figure 3.7. SDSS galaxy sample *data set. Histograms of (left)* color(u-r)*: ranked 1 in our method, (right)* logMass*: ranked 1 in SURFING among 1-D subspaces.*

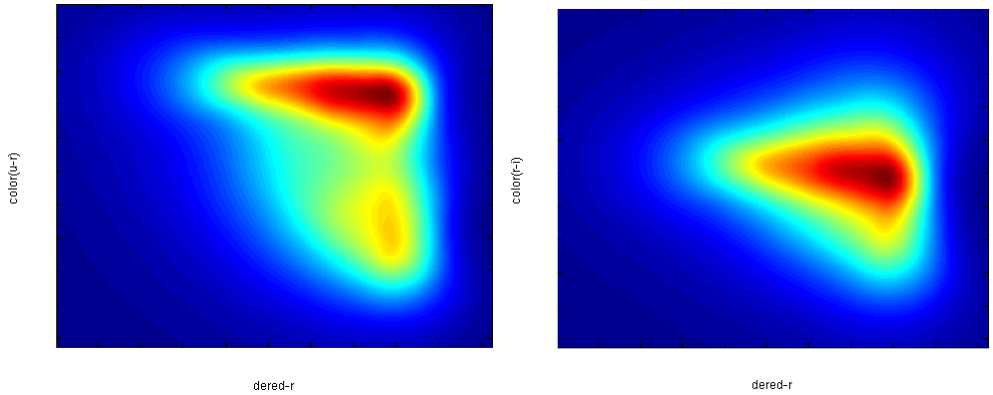

Figure 3.8. SDSS galaxy sample *data set. Color vs Magnitude relation. Left: ranked 1 in our method:* dered_r *vs* color(u-r). *Right: ranked 1 in SURFING:* dered_r *vs* color(r-i).

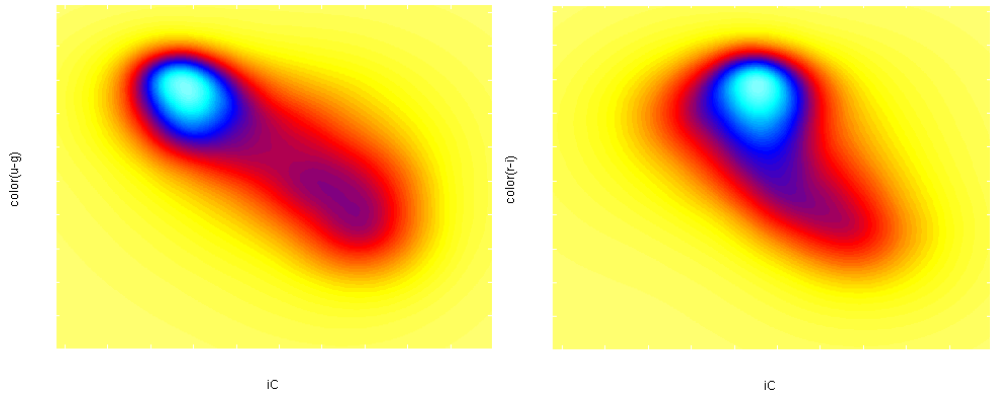

Figure 3.9. SDSS galaxy sample *data set. Color vs inverse Concentration index relation. Left: ranked 1 in our method:* iC *vs* color(u-g). *Right: ranked 1 in SURFING:* iC *vs* color(r-i).

Table 3.3. *Attributes used in* SDSS galaxy sample

Attribute Name	**Description**
dered_r	Extinction corrected model-magnitude in the r-band.
10 colors: **u-g**, **u-r**, **u-i**, **u-z**, **g-r**, **g-i**, **g-z**, **r-i**, **r-z**, **i-z**	A quantitative measure of color of a galaxy is defined as the difference between magnitudes at two different effective wavelengths
logMass	Mass of the galaxy (in logarithmic scale)
logDensity	Number density of galaxies of the environment surrounding the galaxy (in logarithmic scale)
iC	Inverse Concentration index, a measure for the structure of the galaxy
SBr	Surface brightness of the galaxy

plot of *dered_r* vs *color(u-r)* subspace, see figure 3.8, whereas virtually no bimodality can be seen in the *dered_r* vs *color(r-i)* subspace. Similar observations hold for the *color* vs *iC* relation. Here we also found that the bimodality is best visible in the subspace chosen by our method, see Fig. 3.9. The performance of our method remains the same in higher dimensions, see Figure 3.10, where our method shows its strength in detecting relevant subspaces.

The performance of CLIQUE on the galaxy dataset is poor. We experimented with various parameter settings but could not find any of the known galaxy relations we were looking for.

Computation time. For synthetic dataset 1 (5000 data points) it took 0.001s, 1.15s, and 7.75s for computing the 1D, 2D, and 3D density field, respectively, while for the *Galactic stellar halo* dataset (with 3.3 million data points) it took 1.52s, 3.6s, and 217.72s. For both datasets an automatic choice of the smoothing parameter was used.

3.6 Gene Expression Data

We used the gene expression data of the budding yeast Saccharomyces Cerevisiae described by Spellman *et al.* (1998). We downloaded the dataset from the website `http://genome-www.stanford.edu/clustering`, which is an on-line supplement to the paper of Eisen et al. (Eisen *et al.* 1998). Each cell of the data table represents the measured $Cy5/Cy3$ fluorescence ratio ($\log_2$-transformed) at the corresponding target element on the appropriate microarray. Genes are mapped to rows, time points to columns of the data table. That is, each row in the data table contains the time profile of the expression of a particular gene. Each column represents the expression values of all genes at a particular time point at which the array experiment was carried out.

Eisen *et al.* (1998) presented a visualization of this gene expression dataset using hierarchical clustering. They devised a similarity measure between the time profiles as the basis for the clustering and visualized the corresponding dendrogram along with the table where each cell value is transformed into a color (red/green/black). With such a visualization it is possible to gain insight in clusters formed by groups of genes with similar expression patterns. However, as they applied

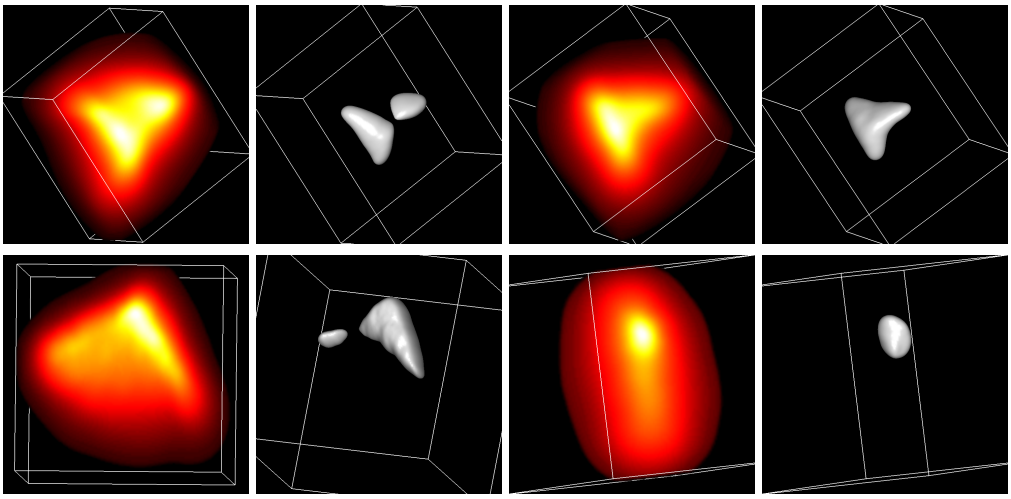

Figure 3.10. *Visualization of* SDSS galaxy sample *dataset. Row 1: Volume visualization of 3-D subspaces. From left to right: ranked 1 in our method:* dered_r *vs* color(u-r) *vs* SBr *(Xray and isosurface); ranked 1 in SURFING:* dered_r *vs* color(i-z) *vs* SBr *(Xray and isosurface). Row 2: Volume visualization of first three principal components of 5-D subspaces. From left to right: ranked 1 in our method:* dered_r *vs* color(u-i) *vs* color(i-z) *vs* iC *vs* logMass *(Xray and isosurface); ranked 1 in SURFING:* color(g-r) *vs* color(g-z) *vs* color(r-i) *vs* color(i-z) *(Xray and isosurface).*

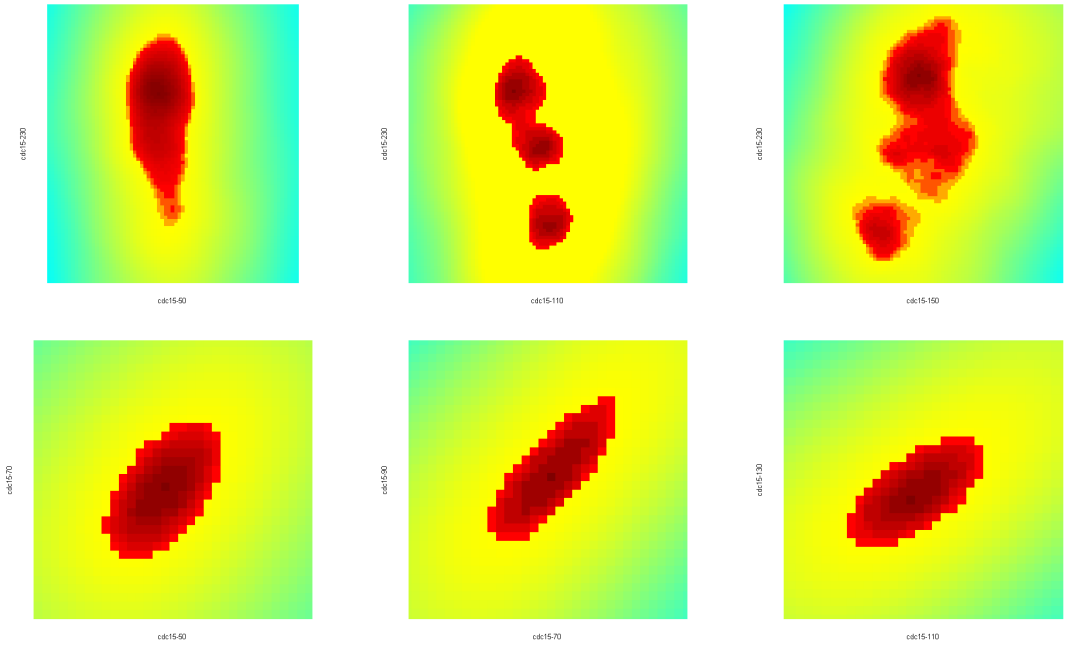

Figure 3.11. *Density images (zoomed for better observation) of subspace ranking for the gene expression dataset.* Top row*: the three highest ranking 2D subspaces selected by our method; from left to right: subspace (cdc*15_50, *cdc*15_230*) with two clusters; (cdc*15_110, *cdc*15_230*) with three clusters; and (cdc*15_150, *cdc*15_230*) with five clusters.* Bottom row*: the three highest ranking 2D subspaces selected by SURFING; from left to right: (cdc*15_50, *cdc*15_70*), (cdc*15_70, *cdc*15_90*), and (cdc*15_110, *cdc*15_130*), all of these subspaces showing a single cluster.*

a full-dimensional clustering technique, it is possible to miss some of the clusterings/groupings of genes that cannot be seen in the full-dimensional space. Further insights and new relations among genes can be explored using our subspace clustering approach.

Similar to Baumgartner *et al.* (2004), we only used the part of the data table that refers to the cell-cycle, i.e., the *cdc*15 "arrest and release" experiment. This means that the cells were first synchronized by growing them at a very high temperature until most of them had reached a similar state of arrest characterized by large dumbbells. Then the cells were released from the arrest by shifting the culture to a 23 °C water bath, and samples were taken starting at $t = 10$ min with 20 min intervals (Spellman *et al.* 1998). The cell cycle was monitored by the appearance of new buds. During the cell cycle, first new buds appeared at $t = 50$ min, $t = 150$ min, and $t = 270$ min, meaning that the cells completed slightly more than two full cycles during the experiment. From the figures in Spellman *et al.* (1998) it can be observed that the expression patterns vary periodically, with different groups of genes being co-expressed at different phases during the cell cycle, although the first and second cycle are not of the same length.

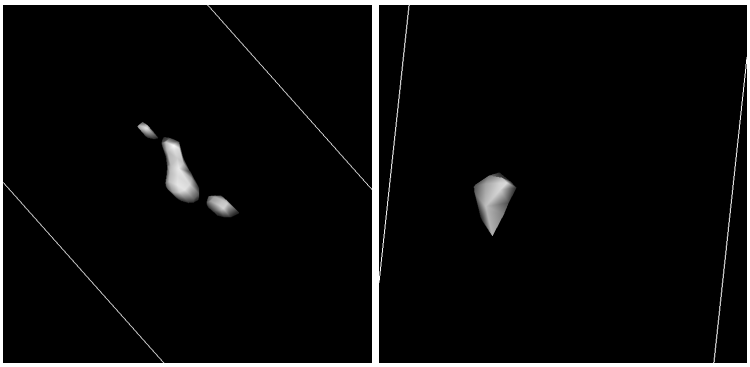

Figure 3.12. *Subspace ranking for the gene expression dataset.* Left*: highest ranking 9D subspace selected by our method: (cdc*15_10*, cdc*15_70*, cdc*15_90*, cdc*15_110*, cdc*15_130*, cdc*15_150*, cdc*15_210*, cdc*15_230*, cdc*15_270*) (isosurface rendering of the first three principal components, with isovalue* 3845*).* Right*: highest ranking 9D subspace selected by SURFING: (cdc*15_10*, cdc*15_30*, cdc*15_50*, cdc*15_70*, cdc*15_90*, cdc*15_110*, cdc*15_130*, cdc*15_150*, cdc*15_190*) (same isovalue).*

We removed genes with missing cell values. The resulting dataset has 1938 genes (rows) and 15 time points (columns) of the *cdc*15 experiment. The task is to find interesting subspaces of the multidimensional space spanned by the time points, i.e., sets of times at which the expression of particular sets of genes shows strong signs of clustering. Such clustering is a strong indication of co-regulation of genes during the cell cycle.

In the top row of figure 3.11, density images of the three highest ranking 2D subspaces selected by our method are shown. The sample times are used as experiment identifier, for example, *cdc*15_50 denotes the sample of the *cdc*15 experiment at time $t = 50$, etc. In the subspace (*cdc*15_50, *cdc*15_230) a bi-modality can be seen, with one large and one small cluster. In the subspace (*cdc*15_110, *cdc*15_230) we can observe three clusters, two linked and one separate cluster. Five clusters can be observed in the subspace (*cdc*15_150, *cdc*15_230), four linked clusters and one separate cluster. From the observation of these three density images

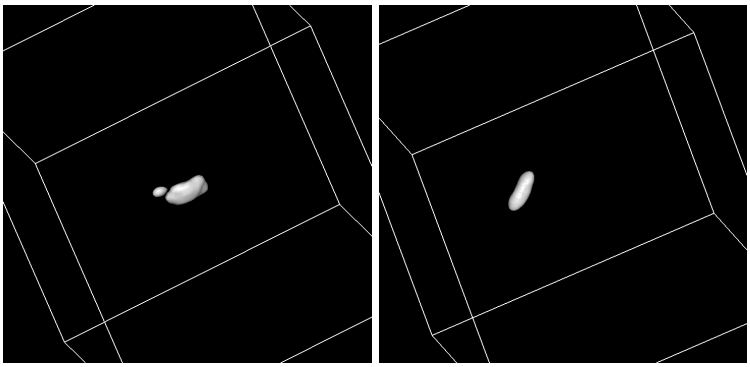

Figure 3.13. *Isosurface rendering of (left) the highest ranking 3D subspace (cdc*15_90, *cdc*15_230, *cdc*15_250*) found by our method; (right) the subspace (cdc*15_190, *cdc*15_270, *cdc*15_290*) found by SURFING.*

we note that our method selects subspaces where the time points refer to different cell cycles during the experiment. Also, we find that the number of clusters increases in later stages of the experiment. This could be attributed to the fact that the synchronization of the cells became less tight towards the end of the experiment (Spellman *et al.* 1998).

On the other hand, the bottom row of figure 3.11 shows that the subspaces (*cdc*15_50, *cdc*15_70), (*cdc*15_70, *cdc*15_90), and (*cdc*15_110, *cdc*15_130) chosen by SURFING display a unimodal distribution and all of the subspaces refer to a single cell cycle during the experiment. In the 1D ranking of SURFING, *cdc*15_230 (one of the highest ranking 1D subspaces in our method) gets the lowest quality value, which is even lower than ($\frac{2}{3}\cdot$ quality of highest ranked 1D subspace). According to SURFING's pruning criteria this subspace is removed as irrelevant from the set of subspaces, see the algorithm in (Baumgartner *et al.* 2004, Fig. 3).

In figure 3.12 an isosurface rendering of the first three principal components of two 9D subspaces can be seen. Subspace (*cdc*15_10, *cdc*15_70, *cdc*15_90, *cdc*15_110, *cdc*15_130, *cdc*15_150, *cdc*15_210, *cdc*15_230, *cdc*15_270) is the highest ranking 9D subspace in our method (figure 3.12, left). Three dense clusters are visible in the isosurface rendering of the subspace with isovalue 3845. On the other hand, in the rendering (with the same isovalue) of the highest ranking 9D subspace (*cdc*15_10, *cdc*15_30, *cdc*15_50, *cdc*15_70, *cdc*15_90, *cdc*15_110, *cdc*15_130, *cdc*15_150, *cdc*15_190) chosen by SURFING, only one cluster is visible (figure 3.12, right).

Baumgartner *et al.* reported a number of significant clusters using the SURFING method for the gene expression data set studying the yeast mitotic cell cycle (Table 2 of Baumgartner *et al.* (2004)). One cluster is formed by a 3D subspace comprising time points 190, 270, and 290 (shown in the right of figure 3.13). They also found two 4D subspaces (bottom of figure 3.14), one subspace with time points 90, 110, 130, and 190 with three clusters and another one with time points 70, 90, 110, and 130 with three clusters. The latter one is among the top 10 subspaces found in our implementation of SURFING. Though the 3D subspace and other 4D subspace are not in the top 10 in our implementation, they have quality values greater than the threshold quality value.

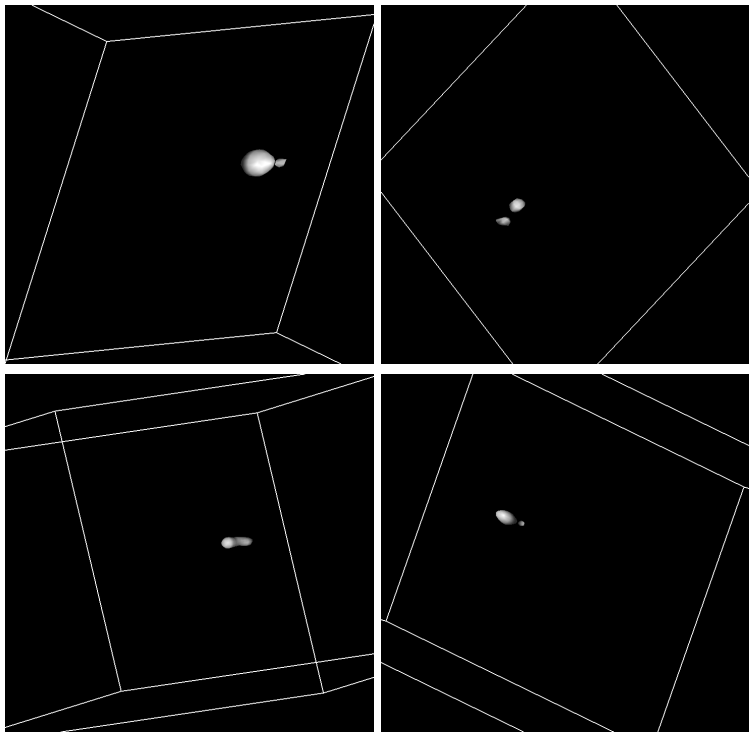

Figure 3.14. *Isosurface rendering of the first three principal components of (top left and right) the two top ranked 4D subspaces (cdc*15_70, *cdc*15_130, *cdc*15_190, *cdc*15_210*) and (cdc*15_190, *cdc*15_210, *cdc*15_250, *cdc*15_270*) chosen by our method; (bottom left and right) the two subspaces (cdc*15_70, *cdc*15_90, *cdc*15_110, *cdc*15_130*) and (cdc*15_90, *cdc*15_110, *cdc*15_130, *cdc*15_190*) found by SURFING.*

3.7 Summary and Future Plans

In this chapter, we have presented a method for ranking subspaces in high-dimensional data in terms of their relevance for clustering. We used connected morphological operators on a grid-based density field that provides not only a good quality criterion but also has visual support for the analysis of the subspaces. Evaluation of the method on synthetic, astronomical and gene expression datasets confirmed its strength in finding relevant subspaces and the usefulness of its visualization. In our approach we allow the user to interact with the system even during the search process, and directly confirm the results by looking into the density image produced. Our interactive application where tree visualization has been integrated with well-established visualization tools aids the user to achieve further in-depth knowledge by exploration of the subspaces.

Future work will concern further improvement of the results using dynamics-based filtering of the density image. Our quality criterion could also be used to find an optimal smoothing parameter. We also will investigate extension of the Max-tree algorithm to dimension higher than three. This would enable subspace ranking without recourse to PCA in higher dimension. This however would also require the use of visualization techniques in dimension higher than three. Several methods are available for this purpose, such as parallel coordinate plots (Inselberg 2009), scatter plot matrices (Chambers *et al.* 1983), or tours (Asimov 1985, Cook *et al.* 1995, Feigelson and Babu 2003). However, parallel coordinate plots are less intuitive and it is hard to discern structures, especially if the dataset is very large. It can be even difficult to find correlations as these are sometimes misinterpreted in parallel coordinate plots (Li *et al.* 2010). Scatter plot matrices can be useful to see pairwise relationships of features, but being two dimensional in nature three dimensional structures cannot be seen.

We are currently integrating our approach with a multi-touch display system. This will allow scientists to discuss their results in a collaborative environment which supports both scientific and information visualization. A user evaluation of the complete system will be carried out.

Acknowledgments

We thank prof. Amina Helmi of the Kapteyn Astronomical Institute for the Galactic stellar halo dataset used in this chapter.